BRIEF COMMUNICATION

On Searching Misspelled Collections

Jason Soo
Information Retrieval Laboratory, Georgetown University, 3700 O Street NW, Washington, DC 20057.
E-mail: soo@ir.cs.georgetown.edu

Ophir Frieder
Information Retrieval Laboratory, Georgetown University, 3700 O Street NW, Washington, DC 20057.
E-mail: ophir@ir.cs.georgetown.edu

We describe an unsupervised, language-independent spelling correction search system. We compare the proposed approach with unsupervised and supervised algorithms. The described approach consistently outperforms other unsupervised efforts and nearly matches the performance of a current state-of-the-art supervised approach.

Introduction

Over two thirds of misspelled queries are caused by transformation errors (insertion, deletion, replacement, and inversion; Li, Duan, & Zhai, 2012; Pollock & Zamora, 1984). Spelling-correction approaches must address these common transformation errors but many cannot without training data. For example, the USHMM[1] has a document collection comprising 13 languages. The collection is too large for the low volume of queries to be used for training. Worse, should a supervised approach be deployed, the model might overfit to frequently queried languages, biasing against the results for minority languages.

Although there are many kinds of spelling-correction algorithms, three prominent types are:

- Supervised approaches: Recent supervised approaches (Li et al., 2012) are quite powerful. However, when training data are unavailable or incomplete, a supervised approach cannot be used. For complete evaluation, we compare the effectiveness of this recent work with our own, although the approaches are designed for different environments.
- Unsupervised approaches: Earlier efforts (Aljlayl & Frieder, 2002; Aqeel, Beitzel, Jensen, Grossman, & Frieder, 2006; Gey & Oard, 2001) demonstrated that efficient, simple, unsupervised, rules-based approaches are quite effective. More recently advanced n-gram solutions were demonstrated as only somewhat inferior to supervised approaches (Duan & Hsu, 2011) in particular cases. However, n-grams consider only substrings of sequential characters, and rules-based approaches do not leverage the power of n-grams. We build on the findings of this research, correcting for these limitations by crafting a rule-based approach and using n-grams should our confidence be low.
- Phonetic approaches: Soundex (Mokotoff, 2007) and D-M Soundex are two seminal phonetic algorithms that group like-sounding characters into bins. However, their encoding schemes introduce several problems: dependence on initial letter (Patman & Shaefer, 2003), noise intolerance (Patman & Shaefer, 2003), poor precision (Mitton, 1996; Snae & Bruckner, 2009), and ignoring consonants (Beider & Morse, 2008).

In general, supervised algorithms outperform unsupervised algorithms, particularly in cases in which context is important in correcting a word (Lim, 2012); however, they cannot be used in the absence of training data. We describe an unsupervised approach that has no dependence on domain, language structure, or sequential windows (Soo, 2013; Soo & Frieder, 2010). The proposed solution outperforms prior unsupervised solutions and is comparable with a leading supervised approach.

Approach

Our proposed solution, shown in Figure 1, is an unsupervised algorithm that works as follows:

The user submits a query, q, with terms $t_1 \ldots t_n$, thus $q = t_1 \ldots t_n$. If $t_i \in I$, where I is an inverted index, we move

[1]United States Holocaust Memorial Museum.

Received October 9, 2013; revised January 7, 2014; accepted January 8, 2014

 DOI: 10.1002/asi.23240

```
q' ⟵ ∅;
for t_i ∈ q do
    if t_i ∈ I then
        q'[i] ⟵ t_i;
        next;
    end
    for j ∈ R do
        C_j^i = ∅;
        k = 0;
        while k < K do
            C_j^i ⟵ C_j^i ∪ r_k^j(t_i);
            k ⟵ k++;
        end
    end
    C^i ⟵ ⋃_{j=1}^{R} {C_j^i};
    T_s ⟵ max_{t'∈C^i} {v_t' / Σ^{|C^i|} v_t'};
    if T_s < β then
        N^i ⟵ produce n-grams candidates;
        T_n ⟵ n-grams confidence;
        if T_s < T_n then
            C^i ⟵ N^i
        end
    end
    q'[i] ⟵ max_{t'∈C^i} {v_t'};
end
return q';
```

FIG. 1. Segments algorithm.

to the next $tj \in q$, where $j \neq i$. Otherwise, $t_i \notin I \Rightarrow t_i$ is misspelled or nonexistent. Assuming the former, and a correct spelling of t_i, $t'_i \in I$, we use a set, currently six, of substring-generation rules, R, to generate candidate sets C_j^i for each t_i: $C_j^i = \mathrm{r_k}^j(t_i)$, where j is the rule number and k is the iteration number. Each C_j^i will contain multiple tuples of the form $(t', \mathrm{v_t}')$, where t' is a candidate correction for t_i and v_t' counts the number of times r_k^j found t'. Rules 1 and 2 (described in the next section) execute for K iterations or until the substring has fewer than four characters. In our testing, $K = 4$. Increasing K decreases precision and increases the number of unique candidates (akin to recall), because new iterations search for more relaxed matches to the original user query. Evidence of this is shown in Figure 2. Once all the rules for t_i have completed, we compute a global candidate set for t_i as follows: $C^i = \bigcup^R{}_{j=1} \{C_j{}^i\}$.

A confidence T of our t'_i is then derived by dividing the number of votes for it (v_t') by the total number of votes cast. T is measured against a user-defined threshold, β. If $T < \beta$, an n-gram-based solution is deployed. We empirically observed $\beta = 0.3$ to perform best. That is, in addition to the segment search already performed, a traditional n-gram search in which $n = 3$ is conducted, and a confidence for the n-gram solution is computed in the same fashion as T. A comparison of the confidence of the described approach and the n-gram solution is made, and the candidate set with the higher confidence is selected.

At this point, we have selected our most confident candidate sets, Ci, for each term t_i from the user's original query q. We now select the most confident candidates $t_i \in C^i$, where confidence is measured in the same way as the previously described T. The result is our most confident permutation of candidate strings, $q = t_1 \ldots t_n$.

Substring-Generation Rules

We studied many generation rules and experimentally found the following six to be the most suitable. The primary contributions from these rules are, (a) unlike n-gram solutions, they generate candidates based on nonsequential character windows, which increases the correction rate of our approach, and (b) they have no dependence on language or domain.

For simplicity of illustration, in the following rule definitions, $\alpha(x)$ searches against the index I, creates the associated candidate sets C^i, and returns the input term x. Furthermore, we forego illustrating verification of string length to prevent runtime errors. The percentage symbol (%) represents a wildcard indicating zero or more arbitrary characters. For illustration purposes, the derived substrings for the search term *Mississippi* are given in Table 1.

$$\mathrm{r_k}^1(\mathrm{t_i}) = \begin{cases} r_{k+1}{}^1(\alpha(\%\, t_i[2..n-1]\,\%)) & \text{if } k < K \\ \alpha(\%\, t_i[2..n-1]\,\%) & \text{if } k = K \end{cases}$$

$$\mathrm{r_k}^2(\mathrm{t_i}) = \begin{cases} r_{k+1}{}^2(\alpha(t_i[1..\lfloor (n/2)-1 \rfloor]\,\%\, t_i[\lfloor (n/2)+1 \rfloor..n])) & \text{if } k < K \\ \alpha(t_i[1..\lfloor (n/2)-1 \rfloor]\,\%\, t_i[\lfloor (n/2)+1 \rfloor..n]) & \text{if } k = K \end{cases}$$

$$\mathrm{r_k}^3(\mathrm{t_i}) = \alpha(\%\, t_i[\lfloor (n/2)+1 \rfloor..n])$$

$$\mathrm{r_k}^4(\mathrm{t_i}) = \alpha(t_i[1..\lfloor (n/2)-1 \rfloor]\%)$$

$$\mathrm{r_k}^5(\mathrm{t_i}) = \alpha(t_i[1]\,\%\, t_i[n])$$

$$\mathrm{r_k}^6(\mathrm{t_i}) = \alpha(t_i[1..2]\,\%\, t_i[n-1..n])$$

Evaluation

We evaluated the proposed approach over four data sets, specifically:

- Web Query Logs, a subset of the Microsoft Research annotated TREC 2008 Million Query Track data set[2] used by Li et al. (2012). This collection supports comparison with their work and evaluates in context-aware situations. Our selection processes has been outlined further by Soo (2013).
- USHMM Names, selected names from the Yizkor book collections and victim records (Amir, 2001). It is a collection of 13 different languages, allowing evaluation of the proposed approach on different and diverse languages.

[2]http://research.microsoft.com/en-us/downloads/ff7aba09-fbb4-4201-bc98-23e2a3674e3c/

DOI: 10.1002/asi

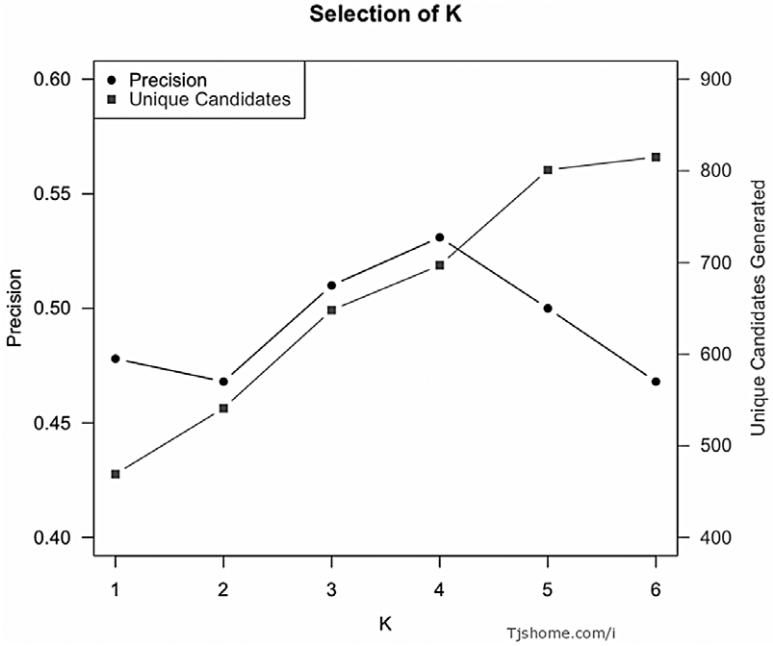

FIG. 2. K is directly tied to precision and recall.

TABLE 1. Substring-generation rules applied to Mississippi. Each column is an rule iteration.

Rule	Search candidates		
1	%ississipp%	%ssissip%	%sissi%
2	Missi%sippi	Miss%ippi	Mis%ppi
3	%ssippi		
4	Missis%		
5	M%i		
6	Mi%pi		
k	*k* = 1	*k* = 2	*k* = 3

TABLE 2. Data set statistics.

		Query stats				
Name	Size	Mean	Min	Max	Median	Mode
USHMM Spoken Names	249	9.80	5	28	9	8
USHMM Names	656	63.77	7	164	54	60
USCB	88,799	7.83	3	14	8	7
Web Query Log	93	29.98	6	60	27	46

- USHMM Spoken Names Query Log,[3] foreign name user query logs. Unlike other data sets, which are synthetically altered (USHMM Names and USCB Names), these contain errors generated by users.
- USCB Names, a collection of names, sorted by frequency, according to the results of the U.S. Census Bureau (USCB) during their 1990 collection.[4] This evaluates the proposed approach on a reasonably large name collection.

Statistical descriptions of each data set can be found in Table 2.

Query Generation

Since the USHMM Names and USCB Names data sets do not contain, at least by definition, misspellings, we synthetically generated them. That is, we used four functions that, given a query and a magnitude, return the query with a transformation error of the magnitude 1–4. For example, passing a query *jason* and a magnitude of 2 to the insertion function may return *jXasonL*. All indexes and characters manipulated are randomly selected. No checks are done to determine whether the resulting query is a valid term. However, if a valid term is generated, all evaluated approaches will be equally affected, because all approaches are modified to first search for an exact match. Such instances were not observed.

Methods

Each data set contains multiple tuples comprising a misspelled query and a target query. For all of our experiments, we provide the misspelled query to each approach, and then record whether the target query was returned and in what ranked position. If the misspelled query was synthetically generated from the target query, we run the experiment three times, reporting the average of the runs. We then report our findings as follows.

[3]http://cs.georgetown.edu/~jason/data/yizkor

[4]http://www.census.gov/genealogy/www/data/1990surnames/dist.all.last

DOI: 10.1002/asi

Web Query Logs results are reported by F1score for comparison with prior work. The remaining results are reported using three metrics:

- Percent found: If, given a misspelling, the correct query was returned in the top 60 results, the query was found. Percentage found is the ratio (number of queries found)/(total queries attempted).
- Rank: The average rank of the target term within the rank-sorted candidate set returned from each query.
- Common rank: The average rank evaluated strictly on the subset in which both n-grams and the proposed approach find the target term. This measure compares the average rank of the proposed approach (often worse overall because of it finding harder terms) with the average rank of 3-grams.

Results

The Web Query Log results presented in Table 3 illustrate F1 scores at three different levels: *All* shows the score for all transcription error queries; *CF* and *CD* represent the scores for context-free (84%) and context-dependent (16%) queries, respectively. Context-dependent queries incorporate surrounding words for corrections. For example, a query *capital* is correct, but *capital hill* is not (*capitol hill* would be correct). The former represents CF and the latter CD. Intuitively, the learning approach does outperform the proposed approach. Interestingly, though, it only has an overall F1 score of 0.005 higher. Furthermore, the proposed approach does considerably better with CF corrections than CD ones.

Table 4 displays the results from our USHMM Spoken Names Query Log evaluation. In all measurements, the proposed approach shows statistically significant ($p < 0.01$) performance increases and never has worse results than n-grams.

Table 5 shows results from the USCB and USHMM Names data sets. The data sets are separated by a column with a double line. We measure the percentage found (%) and rank for both 3-grams (3G) and the proposed approach (US). The rank columns have comma-separated values of rank and common rank. Each row represents the average of three tests for the specified query generation function. The result of the better performing approach is italicized. The proposed approach has the best results in all but three instances in which the 3-grams solution supports a slightly better common rank than the proposed approach.

TABLE 3. Web query log results.

Algorithm	All	CF	CD
US	0.526	0.564	0.333
Li's Approach	0.531	N/A	N/A

TABLE 4. USHMM spoken names query log.

	Perxentage found	Average rank	Common rank
US	78.19	8.26	7.39
3-Grams	62.17	12.00	N/A

Conclusions

Greater than half of all spelling errors today consist of transformation errors. Supervised approaches handle these problems well. However, when training data are insufficient,

TABLE 5. USCB and USHMM names results.

	USCB results				USHMM Names results			
	3G (%)	3G rank	US (%)	US rank	3G (%)	3G rank	US (%)	US rank
INS								
1 Char	99.57	1.59, 1.58	**99.86**	1.37, **1.35**	94.94	2.55	**100**	**1.71**, **1.71**
2 Char	96.0	3.01, 2.76	**97.17**	2.33, **2.14**	91.72	3.45	**99.32**	2.61, **2.43**
3 Char	90.45	4.43, 3.64	**91.83**	2.85, **2.35**	87.82	4.11	**97.52**	3.18, **3.02**
4 Char	84.30	5.71, 4.19	**85.85**	3.34, **2.49**	83.85	5.00	**95.23**	3.87, **3.79**
DEL								
1 Char	98.29	2.97, 2.90	**99.1**	2.64, **2.54**	93.33	3.45	**99.97**	**2.51**, 2.53
2 Char	86.12	6.3, 4.66	**86.86**	5.82, **4.60**	84.87	4.81	**97.96**	4.72, **3.95**
3 Char	70.84	8.72, **5.33**	**70.91**	8.67, 5.83	74.68	5.77	**92.71**	6.42, **4.84**
4 Char	53.73	8.48, **4.12**	**56.28**	10.24, 5.73	70.31	5.95	**86.51**	7.12, **5.14**
REP								
1 Char	96.97	2.63, 2.46	**98.93**	2.29, **2.16**	92.33	3.27	**100**	2.15, **2.01**
2 Char	80.79	5.16, 3.81	**85.5**	4.92, **3.71**	80.95	4.49	**93.90**	4.19, **3.31**
3 Char	62.77	6.95, **4.19**	**68.38**	6.55, 4.21	69.28	5.20	**85.61**	5.60, **3.87**
4 Char	47.29	8.40, 4.321	**52.75**	7.74, **4.315**	57.81	5.98	**75.18**	6.85, **4.84**
INV								
Adj.	87.22	5.39, 4.33	**92.32**	4.81, **3.98**	84.89	4.88	**98.00**	3.77, **3.00**
2 Char	40.71	10.39, 4.87	**44.47**	9.55, **4.56**	54.59	6.78	**71.61**	7.38, **5.39**
3 Char	30.90	12.22, 5.36	**33.44**	11.38, **5.36**	42.89	7.40	**57.59**	8.55, **6.22**
4 Char	22.53	13.44, 4.78	**25.1**	12.25, **4.68**	34.64	8.49	**46.76**	9.29, **7.26**

DOI: 10.1002/asi

supervised approaches fail. We propose an unsupervised search approach in misspelled environments that has no dependence on language structure. We evaluated this against phonetic, supervised, and unsupervised approaches and found favorable results. It has the highest percentage found compared with phonetic and unsupervised approaches and has an F1 score within 0.005 of a modern supervised algorithm. Furthermore, it almost always yields the most favorable rank. We demonstrate the ability of the proposed approach to operate on collections that contain hundreds of thousands of records, with only a scarce few cases in which the rank is negatively affected. The approach described is in use within the archives section of the United States Holocaust Memorial Museum.

References

Aljlayl, M., & Frieder, O. (2002). On arabic search: Improving the retrieval effectiveness via a light stemming approach. In C. Nicholas (Ed.), Proceedings of the Eleventh International Conference on Information and Knowledge Management (ACM CIKM '02) (pp. 340–347). Mclean, VA: ACM.

Amir, M. (2001). From memorials to invaluable historical documentation: Using yizkor books as resource for studying a vanished world. Paper presented at Annual Convention of the Association of Jewish Libraries, La Jolla, California.

Aqeel, S., Beitzel, S., Jensen, E., Grossman, D., & Frieder, O. (2006). On the development of name search techniques for arabic. Journal of the American Society of Information Science and Technology, 57(6), 728–739.

Beider, A., & Morse, S. (2008). Beidermorse phonetic matching: An alternative to soundex with fewer false hits. Avotaynu: The International Review of Jewish Genealogy, 24(2), 12–25.

Duan, H., & Hsu, B. (2011). Online spelling correction for query completion. In S. Sadagopan, K. Ramamritham, A. Kumar, & M.P. Ravindra (Eds.), Proceedings of the 20th International Conference on World Wide Web (ACM WWW '11) (pp. 117–126). Hyderabad: ACM.

Gey, F., & Oard, D. (2001). The trec-2001 cross-language information retrieval track: Searching Arabic using English, French or Arabic queries. Paper presented at The Tenth Text Retrieval Conference, Gaithersburg, MD.

Li, Y., Duan, H., & Zhai, C. (2012). A generalized hidden markov model with discriminative training for query spelling correction. In W. Hersh (Ed.), Proceedings of the 35th International Conference on Research and Development in Information Retrieval (ACM SIGIR '12) (pp. 611–620). Portland, OR: ACM.

Lim, Y. (2012). Spell checking powered by the web. In Google docs blog. Retrieved from http://googledrive.blogspot.com/2012/03/spell-checking-powered-by-web.html

Mitton, R. (1996). Spellchecking by computers. Journal of the Simplified Spelling Society, 20(1), 4–11.

Mokotoff, G. (2007). Soundexing and genealogy. Retrieved from http://www.avotaynu.com/soundex.html

Patman, F., & Shaefer, L. (2003). Is soundex good enough for you? The hidden risks of soundex-based name searching. Herndon, VA: Language Analysis Systems, Inc.

Pollock, J., & Zamora, A. (1984). Automatic spelling correction in scientific and scholarly text. Communications of the ACM, 27(4), 358–368.

Snae, C., & Bruckner, M. (2009). Novel phonetic name matching algorithm with a statistical ontology for analysing names given in accordance with thai astrology. Issues in Informing Science and Information Technology, 6, 497–515.

Soo, J. (2013). A non-learning approach to spelling correction in web queries. In Proceedings of the 22nd International Conference on World Wide Web (ACM WWW '13) (pp. 101–102). Rio de Janeiro: ACM.

Soo, J., & Frieder, O. (2010). On foreign name search. In C. Gurrin, Y. He, & G. Kazai (Eds.), European Conference on IR Research (ECIR '10) (pp. 483–494). Milton Keynes, UK: Springer.

DOI: 10.1002/asi